



RJ
PD990062
26/8

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Marco Winter and Harald Schiller
Serial No.: 09/691,806
Filed: October 19, 2000
For: Method for Coding a Picture Sequence or Sub Picture Unit
Examiner: Huy Thanh Hguyen
Art Group: 2616
Customer No.: 24498

LETTER ENCLOSING COPY OF PRIORITY DOCUMENT

Hon. Commissioner for Patents
Alexandria, VA 22313

Sir:

In accordance with the attached communication dated June 17, 2005, from Examiner Nguyen, attached herewith is a copy of the foreign priority document for the above-identified application.

As noted by the Examiner, the previous communication mailed June 17, 2005 indicated that a foreign priority document had not been received by the Patent Office. Enclosed herewith is a copy of that priority document.

Respectfully submitted,

MARCO WINTER, ET AL.

By: Jorge Tony Villabon
Jorge Tony Villabon
Registration No. 52,322
Tel: 609-734-6445

Date: 8/8/05

Patent Operation
Thomson Licensing
P.O. Box 5312
Princeton, NJ 08543-5312

CERTIFICATE OF MAILING CFR 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

8/9/05

Date

Linda Tindall
Linda Tindall

PDF Acct

BUNDESREPUBLIK DEUTSCHLAND



Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

CERTIFIED COPY OF PRIORITY DOCUMENT

Aktenzeichen: 199 50 490.3

Anmeldetag: 20. Oktober 1999

Anmelder/Inhaber: Deutsche Thomson-Brandt GmbH, Villingen-Schwenningen/DE

Bezeichnung: Verfahren zur Kodierung einer Bildsequenz sowie Teilbilddateneinheit zur Verwendung in einem elektronischen Gerät und Datenträger

IPC: H 04 N 5/92

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 17. Oktober 2000
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

A handwritten signature in black ink, appearing to read "Hoß".

Hoß

**Verfahren zur Kodierung einer Bildsequenz sowie
Teilbilddateneinheit zur Verwendung in einem elektronischen
Gerät und Datenträger**

5

Die Erfindung betrifft sowohl ein Verfahren zur Kodierung einer Bildsequenz als auch weiterhin eine dementsprechende Teilbilddateneinheit zur Verwendung in einem elektronischen Gerät. Weiterhin betrifft die Erfindung einen Datenträger, auf 10 dem eine Teilbilddateneinheit gemäß der Erfindung abgespeichert ist.

Stand der Technik

15

Aus der Videotechnik sind verschiedene Bildkodierverfahren bekannt. Das hier vorgestellte Verfahren zur Kodierung einer Bildsequenz lehnt sich an ein Bildkodierverfahren an, das zur Erzeugung von sogenannten Teilbilddateneinheiten (im 20 Englischen "Sub picture-unit" genannt) für DVD-Geräte benutzt wird. Derartige DVD-Geräte sind bereits im Verkauf und enthalten jeweils eine Teilbilddekodiereinrichtung (im Englischen "Sub picture-decoder" genannt), die die erwähnten 25 Teilbilddateneinheiten verarbeiten kann. Eine Beschreibung des dort benutzten Bildkodierverfahrens findet sich z.B. in dem DVD-Standard "DVD Specification for read-only disc", Part 3, Video specification, Version 1.0, August 1996, Kapitel 5.4.3 Sub-picture-unit (SPU). Eine detaillierte Beschreibung der erwähnten Teilbilddekodiereinrichtungen findet sich z.B. in 30 EP-A-0 725 541.

Gemäß dem DVD-Standard, ist der Hauptanwendungsfall für Teilbilddateneinheiten z.B. die Generierung von Untertiteln

auf dem Bildschirm eines Fernsehgerätes. Die Untertitel können dann irgendwo im Bild eingeblendet werden, meistens werden sie am unteren Bildrand eingeblendet. In der Teilbilddateneinheit SPU sind dann die Bit-Map-Daten für jeweils einen Untertitel abgespeichert. Zur Festlegung des Ortes eines Untertitels auf dem Bildschirm sowie auch zur Festlegung einiger Darstellungscharakteristika, insbesondere Farbe, Transparenz, etc. sind in dem genannten DVD-Standard einfache graphische Steuerbefehle spezifiziert. Damit lassen sich bei geschickter Programmierung Teilbilddateneinheiten prinzipiell auch für andere Zwecke einsetzen, so z.B. für Bedienungshinweise, Bedienerführungsmenüs, Animationsbilder, Pausenbilder etc.

Die wesentliche Teile einer Teilbilddateneinheit gemäß dem DVD-Standard bestehen in einem Datenfeld für die Bit-Map-Daten des darzustellenden Textes bzw. allgemeiner des darzustellenden Objektes sowie in einem zweiten Teil, in dem sogenannte Anzeigesteuерungsbefehlssequenzen abgespeichert sind, die den Darstellungszeitpunkt, die Darstellungszeit sowie die Darstellungscharakteristika des abgespeicherten Objektes festlegen. In dem Datenfeld für die Bit-Map-Daten können durchaus auch mehrere Objekte nacheinander abgespeichert sein, die jeweils zu verschiedenen Zeiten aufgerufen werden. Sollen in einer Teilbilddateneinheit mehrere Objekte vorgesehen werden, so gibt der DVD-Standard die Anleitung, jeweils pro Objekt einen entsprechenden Bereich in dem Bit-Map-Datenfeld vorzusehen. Die Bit-Map-Daten sind dann noch lauflängenkodiert um eine Bildkompression zu erreichen und Speicherplatz zu sparen. Während in dem erwähnten DVD-Standard bereits Vorkehrungen getroffen sind, wie mit dem vorhandenen Anzeigesteuерungsbefehlsatz ein Durchlaufen eines Textes in vertikaler Richtung in dem eingeblendetem Teilbildfenster realisiert werden kann, gibt es keine Hinweise, wie ein

Durchlaufen eines Textes in horizontaler Richtung in dem Teilbildfenster realisiert werden kann. Das Durchlaufen eines Textes in horizontaler Richtung wird vielfach auch als Laufschrift bezeichnet und ist bei manchen Fernsehprogrammen häufig zu sehen. Als Beispiel wird das fortwährende Durchlaufen von Börsenkursen bei Fernsehprogrammanbietern, wie z.B. NTV erwähnt.

Möglichkeiten, die sich dem Fachmann noch am ehesten anbieten, solche Laufschriften auch bei DVD-Geräten einzusetzen unter Ausnutzung der spezifizierten Teilbilddateneinheiten sind, daß für jedes einzelne Bild einer solchen Laufschrift entweder eine separate Teilbilddateneinheit abgespeichert wird oder für mehrere Phasen der Laufschrift eine Teilbilddateneinheit vorzusehen wäre, die für die einzelnen Phasen der Bewegung entsprechend viele Bit-Maps enthält, die dann einzeln aufgerufen werden. Diese Lösungen sind ersichtlich sehr speicheraufwendig. Die zweite Lösung kommt aufgrund der Beschränkung der Größe von Teilbilddateneinheiten auf 53220 Bytes sowieso nur für kürzere Laufschriften in Frage.

Erfindung

Es ist Aufgabe der Erfindung, ein Verfahren zur Kodierung einer Bildsequenz anzugeben, das es gestattet basierend auf den im DVD-Standard bereits festgelegten Anzeigesteuerungsbefehlen sowie den Regeln für den Aufbau von Teilbilddateneinheiten eine horizontale Bewegung eines Objektes im Teilbildanzeigefenster speichereffizient zu realisieren. Die gestellte Aufgabe wird durch die Merkmale des unabhängigen Anspruchs 1 gelöst. Der Aufbau einer dementsprechenden Teilbilddateneinheit ist in dem unabhängigen Anspruch 8 angegeben.

Bei dem erfindungsgemäßen Verfahren zur Kodierung einer Bildsequenz wird, das Objektbild durch vertikale Schnitte in eine Anzahl gleich großer Segmente zerlegt. Die Bildpunktdataen 5 von jeweils zwei zusammenhängenden Segmenten werden sodann kombiniert, wobei jeweils aus korrespondierenden Bildpunkten der zu kombinierenden Teile ein kombinierter Bildpunkt erzeugt wird, derart, daß dem kombinierten Bildpunkt ein eindeutiger Typ zugewiesen wird, der der Verteilung der Bildpunktwerte in 10 dem Paar der korrespondierenden Bildpunkte entspricht. Hinzu kommt noch, daß solche kombinierten Bildstücke ausschnittsweise lauflängenkodiert werden. Dies geschieht so, daß für die kombinierten Bildpunkte eines Ausschnitts der Laufschrift/Objektes jeweils eine separate Lauflängenkodierung 15 durchgeführt wird. Die Größe des Ausschnittes korrespondiert dabei zu der Anzahl von Bildpunkten, um die das Objekt pro Bewegungsphase in horizontaler Richtung bewegt werden soll. Also wenn z.B. das Objekt sich pro Bewegungsphase um 8 Pixel von rechts nach links bewegen soll, muß jeweils für alle 20 Zeilen des Objektausschnittes eine separate Lauflängenkodierung jeweils pro 8 Bildpunkte durchgeführt werden. Durch entsprechendes Setzen eines Zeigers auf den jeweiligen Objektausschnitt ist es dann bei der Dekodierung der Bildsequenz einfach möglich, eine Bildsequenz zu erzeugen, die 25 ausgegeben, in einem Fenster auf einem Anzeigegerät für den Betrachter den Eindruck einer horizontalen Bewegung des Objektes relativ zu dem Fenster erweckt. Durch diese spezielle Art der kombinierten Kodierung von benachbarten Segmenten des Objektbildes und durch die spezielle Lauflängenkodierungsart 30 kann der Speicherbedarf für die resultierende Teilbilddateneinheit beträchtlich reduziert werden. Es muß nämlich nicht mehr für jede Phase der Bewegung eine eigene Bit-Map vorgesehen werden sondern nur wesentlich weniger, in etwa

soviiele Bit-Maps wie auch durch vertikale Schnitte erzeugte Segmente vorhanden sind. Zwar können sich bei den kombinierten Objektbildteilen etwas mehr Lauflängenkodewörter ergeben, jedoch ist die Reduzierung an erforderlichen BitMap-Bereichen 5 beträchtlich, so daß auf jeden Fall deutlich weniger Speicherplatz benötigt wird.

Durch die in den abhängigen Ansprüchen aufgeführten Maßnahmen sind vorteilhafte Weiterbildungen und Verbesserungen des im 10 Anspruch 1 angegebenen Verfahrens möglich.

Die Maßnahme, nach der die Größe der Objektsegmente zu der Größe der darzustellenden Objektausschnitte korrespondiert, ist eine Optimierung, durch die die Anzahl von Segmenten auf 15 eine geringes Maß begrenzt wird.

Die Maßnahme, nach der zwei zu kombinierende Bildteile zunächst in vertikaler Richtung um zwei Bildpunkte gegeneinander verschoben werden bevor sie kombiniert werden, 20 stellt sicher, daß immer die richtigen Bildpunkte sich nahtlos aneinanderfügen, wenn nach Verstellung des Zeigers für den Start der Lauflängendekodierung in einer Zeile Bildpunkte des unteren und oberen der beiden kombinierten Bildstücke aufeinanderfolgen. Die Bildqualität ist also optimiert.

25

Die Maßnahme, nach der ein kombiniertes Bildstück gegenüber dem vorhergehenden und nachfolgenden kombinierten Bildstück um die Anzahl Bildpunkte versetzt wird, um die das Objekt pro Schritt in horizontaler Richtung bewegt wird, bringt den 30 Vorteil einer weiteren Reduzierung des Speicheraufwandes. Dies ergibt sich daraus, daß damit nämlich die redundanten Positionen in den einzelnen BitMap-Bereichen eliminiert sind.

Eine erfindungsgemäße Teilbilddateneinheit zur Verwendung in einem elektronischen Gerät, die eine kodierte Bildsequenz enthält welche nach dem erfindungsgemäßen Verfahren zur Kodierung einer Bildsequenz kodiert wurde, hat den

- 5 prinzipiellen Aufbau, daß sie einen Datenbereich für die Bildpunktwerthe der einzelnen Bilder der Bildsequenz enthält und einen Befehlsbereich für Anzeigesteuerungsbefehlsequenzen. Erfindungsgemäß ist dann vorgesehen, daß für die Erzeugung der horizontalen Bewegung des Objektes
- 10 Anzeigesteuerungsbefehlssequenzen vorgesehen sind, die jeweils einen Befehl enthalten, durch den der Start der Laufängendekodierung für ein Bild jeweils um eine Anzahl von Bildpunkten korrespondierend zur gewünschten Bewegung des Objektes versetzt wird. Zu diesem Zweck kann vorteilhaft der
- 15 im DVD-Standard festgelegte Anzeigesteuerungsbefehl SET_DSPXA eingesetzt werden. Ebenfalls ist in den Anzeigesteuerungsbefehlsequenzen zur Reproduktion eines Bildes jeweils ein Befehl vorgesehen, der festlegt auf welche Weise die kombinierten Bildpunkte auszuwerten sind.

20

Zu diesem Zweck können im Fall von DVD-Teilbilddateneinheiten die im DVD-Standard definierten Befehle SET_CONTR oder CHG_COLCON benutzt werden.

25

Schließlich besteht die Erfindung in einem Datenträger, insbesonderer einer DVD-Scheibe, das eine Teilbilddateneinheit gemäß der Erfindung abgespeichert enthält.

30

Zeichnungen

Ausführungsbeispiele der Erfindung sind in den Zeichnungen dargestellt und werden in der nachfolgenden Beschreibung näher erläutert. Es zeigen:

5 Fig. 1 ein DVD-Abspielgerät und ein angeschlossenes Fernsehgerät, das eine eingeblendete Laufschrift darstellt;

10 Fig. 2 ein Beispiel für das bekannte Rahmenformat für die Daten der Teilbilddateneinheit;

Fig. 3 eine Darstellung zur Veranschaulichung wie zwei Objektbildabschnitte miteinander kombiniert werden;

15 Fig. 4 eine lauflängenkodierte Zeile eines Objektbildausschnittes;

20 Fig. 5 eine Illustration wie der SET_DSPXA-Befehl in den Anzeigesteuerbefehlssequenzen auf das erste Lauflängenkodewort verschiedener Ausschnitte zeigt;

Fig. 6 die Kombinierung zweier Bildausschnitte für das Beispiel einer Teilbilddateneinheit gemäß Fig. 7;

25 Fig. 7 ein einfaches Beispiel einer Teilbilddateneinheit gemäß der Erfindung;

Fig. 8 die Kombinierung von Bildstücken einer Laufschrift gemäß einem ersten Ausführungsbeispiel;

30 Fig. 9 die Kombinierung von Bildstücken einer Laufschrift gemäß einem zweiten Ausführungsbeispiel der Erfindung;

Fig. 10 eine Illustration zur Verdeutlichmachung des unterschiedlichen Speicheraufwandes, der erforderlich ist bei den Kombinierungsarten gemäß Fig. 8 und 9 und

- 5 Fig. 11 eine Illustration zur Verdeutlichung wie durch geschickte Kodierung der letzten Zeilen einer Laufschrift zusätzlich Speicherplatz eingespart werden kann.

10

Beschreibung der Erfindung

Die Erfindung wird am Beispiel eines DVD-Gerätes erläutert. In Fig. 1 bezeichnet die Bezugszahl 10 ein herkömmliches TV-Gerät. Mit der Bezugszahl 11 ist ein DVD-Abspielgerät bezeichnet. Dieses ist mit dem TV-Gerät 10 z.B. über ein Scartkabel verbunden. Das DVD-Abspielgerät 11 erzeugt das analoge Fernsehsignal, das in das TV-Gerät eingespeist wird. Mit der Bezugszahl 12 ist ein Anzeigefenster bezeichnet, in dem eine Laufschrift eingeblendet ist. Die Laufschrift kann sich entweder von links nach rechts in dem Anzeigefenster 12 bewegen oder umgekehrt von rechts nach links. Durch die gestrichelte Darstellung wird angedeutet, daß die Laufschrift jeweils nur ausschnittsweise in dem Anzeigefenster angezeigt wird. Die gesamte Laufschrift ist dabei sogar so lang dargestellt, daß sie nicht komplett auf dem Bildschirm dargestellt werden könnte. Gemäß dem DVD-Standard wird in einem Anzeigefenster 12 hauptsächlich ein Untertitel in das Fernsehbild eingeblendet. Dazu wird jeweils eine Teilbilddateneinheit (SPU) pro Untertitel verwendet.

In Fig. 2 ist das bekannte Datenformat einer Bilddateneinheit grob dargestellt. Mit der Bezugszahl 20 ist ein Datenfeld für

ein Kopfteil (SPUH) der Teilbilddateneinheit bezeichnet. Die Bezugszahl 21 bezeichnet ein Datenfeld für die komprimierten Bildelementdaten (PXD) des Teilbildes. Mit der Bezugszahl 22 ist noch ein Datenfeld für eine Anzeigesteuerungsbefehlssequenztabelle (SP_DCSQT) bezeichnet. In dem Datenfeld 22 werden sogenannte Anzeigesteuerungsbefehlssequenzen (SP_DCSQ) abgespeichert. Die Definition der erwähnten Begriffe sowie der verschiedenen Anzeigesteuerungsbefehle ist in dem Eingangs erwähnten DVD-Standard (Version 1.0) angegeben, so daß zur 10 Offenbarung der Erfindung nicht alle Einzelheiten erläutert werden müssen. Es wird aber bezüglich der Offenbarung der Erfindung deshalb ausdrücklich auf diese Veröffentlichung Bezug genommen.

15 Die Bildelementdaten in dem Datenfeld 21 bestimmen das dargestellte Muster des angezeigten Teilbildes. Für jedes Bildelement einer Zeile des Teilbildes wird durch ein 2-Bit breites Datenwort angegeben, ob es sich um ein Hintergrund-Bildelement (Background-Pixel) handelt oder um ein 20 Vordergrund-Bildelement (Pattern-Pixel) oder ob das Bildelement auf eine erste Art und Weise hervorzuheben ist (emphasis-1-pixel) oder auf eine zweite Art und Weise (emphasis-2-pixel). Diese vier Unterscheidungen können mit den zwei Bits getroffen werden. Dabei bedeuten die Binär-Werte im 25 einzelnen:

00 = Hintergrundbildelement
01 = Vordergrundbildelement
10 = Mit Hervorhebung 1 dargestelltes Bildelement
30 11 = Mit Hervorhebung 2 dargestelltes Bildelement.

Zu Berücksichtigen ist dabei, daß die einzelnen Bildelementdaten nicht in dieser rohen Form im Speicher

abgespeichert sind, sondern im komprimierter Form. Hierfür wird eine Lauflängenkodierung durchgeführt. Das Lauflängenkodierungsverfahren ist ebenfalls in dem erwähnten DVD-Standard (1.0) sehr genau erläutert. In dem erwähnten DVD-Standard sind 7 Kodierungsregeln für die Lauflängenkodierung festgelegt. Dabei arbeitet die Lauflängenkodierung so, daß jeweils eine Zeile eines Halbbildes des darzustellenden Teilbildes lauflängenkodiert wird. Die einzelnen Regeln werden hier aber nicht genauer erläutert, da diese aus dem erwähnten DVD-Standard hinreichend bekannt sind. Durch die Lauflängenkodierung wird bei einfachen Bildinhalten eine relativ große Komprimierung erzielt.

Das erfindungsgemäße Verfahren zur Kodierung einer Bildsequenz wird nachfolgend anhand des Beispiels einer Laufschrift erläutert. Selbstverständlich kann mit dem erfindungsgemäßen Verfahren nicht nur eine Laufschrift erzeugt werden sondern es könnte statt dessen ein beliebiges anderes Objekt ausschnittsweise in einem Fenster dargestellt werden so daß in dem Fenster eine Relativbewegung zwischen Objekt und Fenster in horizontaler Richtung erkennbar wäre. Zunächst wird das Objekt, d.h. in diesem Beispiel die gesamte Laufschrift durch vertikale Schnitte zerteilt. Jeder Teil wird dabei gleich groß gewählt. Die Größe eines Teils korrespondiert außerdem zu der Größe des Anzeigefensters, in dem die Laufschrift später eingeblendet werden soll. Dies gilt insbesondere für die Ausdehnung jedes Teils in horizontaler Richtung. Nach der Zerlegung der Laufschrift in Segmente werden dann jeweils zwei aneinandergrenzende Segmente miteinander kombiniert. Dieser Vorgang ist auch in der Figur 10 näher veranschaulicht. Im oberen Teil von Figur 10 ist erkennbar, daß eine Laufschrift in neun Segmente eingeteilt wurde, wobei das neunte Segment nicht mehr die gleiche Größe aufweist wie die vorhergehenden

Segmente und aus diesem Grund künstlich vergrößert wird, in dem es mit Hintergrundbildelementen aufgefüllt wird, so daß ebenfalls wieder ein gleich großes Segment entsteht. Die ersten beiden Bildsegmente werden dann miteinander kombiniert.

- 5 Es entsteht das kombinierte Bildstück K1. Als nächstes werden dann das zweite und dritte Segment miteinander kombiniert, es entsteht die Kombination K2. Wiederum wird als nächstes das dritte und vierte Segment miteinander kombiniert. Es entsteht die Kombination K3, usw., bis schließlich als letzte
- 10 Kombination K8 das achte und neunte Segment der Laufschrift miteinander kombiniert wird.

Die Kombination zweier Objektbildsegmente geschieht dabei so, wie in Figur 3 dargestellt. In der Figur 3 ist dabei in dem

- 15 einen Bildsegment der Buchstabe H enthalten und in dem anderen Bildsegment der Buchstabe A. In der Laufschrift sollen dann diese Buchstaben hintereinander geschrieben sein. Die beiden zu kombinierenden Bildteile sind im linken Teil der Figur 3 übereinander dargestellt. Mit * sind jeweils die Bildpunkte
- 20 des Buchstabens markiert und durch einfache Punkte sind die Bildpunkte des Hintergrundes des jeweiligen Bildsegmentes markiert. Es wird also dabei davon ausgegangen, daß ein Bildpunkt nur zwei Werte annehmen kann, nämlich einmal ob der Bildpunkt zum Leuchten angeregt ist, was dem "An"-Zustand
- 25 entspricht oder nicht was dem "Aus"-Zustand entspricht. Die beiden dargestellten Bildteile bestehen jeweils aus 9 x 8 Bildpunkten. Bei der Kombination dieser Bildstücke wird gedanklich das eine Stück über das andere gelegt, so daß die Bildelemente übereinander liegen. In dem dargestellten
- 30 Beispiel werden die beiden Bildstücke nicht deckungsgleich aufeinandergelegt, sondern um zwei Bildpunkte in vertikaler Richtung gegeneinander verschoben. Dies ist erforderlich, um bei der nachfolgenden Dekodierung der kombinierten Bildstücke

eine zeilenrichtige Ausgabe zu gewährleisten, was nachfolgend noch genauer erläutert wird. Das Übereinanderschieben der Bildabschnitte ist in Figur 3 durch Striche angedeutet. Die Kombination der übereinander gelegten Bildpunkte wird dann 5 nach den folgenden Regeln ausgeführt:

Sind die zu kombinierenden Bildpunkte beide im Zustand "Aus", dann wird im kombinierten Bildstück der kombinierte Bildpunkt an dieser Position der Bildpunkttyp A zugewiesen. Weisen die 10 zu kombinierenden Bildpunkte beide den Wert "An" auf, dann wird dem kombinierten Bildpunkt für diese Position der Bildpunkttyp B zugewiesen.

Hat der untere Bildpunkt den Wert "An" und der darüberliegende 15 Bildpunkt den Wert "Aus", dann wird im kombinierten Bildstück der Bildpunkt an dieser Position mit dem Bildpunkttyp C deklariert. Ist der Wert des unteren Bildpunktes "Aus" und der des darüberliegenden Bildpunktes "An", dann wird dem 20 kombinierten Bildpunkt der Bildpunkttyp D zugewiesen. Das Resultat dieser Bildpunkt kombination ist in Figur 3 rechts oben wiedergegeben.

Im unteren Teil der Figur 3 ist dann angegeben, wie die Zuordnung der vier Bildpunkttypen A - D zu den in dem DVD- 25 Standard festgelegten Bildpunkttypen "Background" b "pattern" p, "emphasis-1" und "emphasis-2" getroffen ist. Es wird aus Bildpunkttyp A ein "Background-pixel", aus Bildpunkttyp B ein "Pattern-pixel", aus Bildpunkttyp C ein "emphasis-1-Pixel" und aus Bildpunkttyp D ein "emphasis-2-Pixel". Die entsprechenden 30 Pixeltypen sind in der Figur 3 unten wiedergegeben. Diese Zuordnung ist im Rahmen der Erfindung nur als Beispiel anzusehen. Es sind im Rahmen der möglichen Permutationen der vier Bildpunkttypen $4! = 24$ verschiedene Zuordnungen möglich.

Das so kombinierte Bildstück ist damit fertig und wird dann noch für die Erzeugung einer DVD-Standard-konformen Teilbilddateneinheit lauflängenkodiert. Erfindungsgemäß wird 5 die Laufenlängenkodierung aber nicht speicherbedarfs- optimierend für eine komplette Zeile des kombinierten Bildstücks durchgeführt, sondern mit einer speziellen Lauflängenkodierungsart bearbeitet, die anhand der Figur 4 näher erläutert wird.

10

Die Figur 4 zeigt ein Beispiel einer Zeile eines kombinierten Bildstückes. Im unteren Teil der Figur 4 sind die Lauflängenkodeworte für die dargestellte Zeile angegeben, für den Fall daß die Lauflängenkodierung speicherbedarfs- 15 optimierend arbeitet. Es ist erkennbar, daß ein Lauflängenkodewort jeweils pro Anzahl von Bildpunkten gleichen Typs (B oder P) gebildet ist. Erfindungsgemäß wird nun aber statt dessen immer eine separate Lauflängenkodierung für acht aufeinanderfolgende Bildpunkte durchgeführt. Durch die 20 gestrichelten Linien im oberen Teil der Figur 4 ist jeweils ein Abschnitt von acht aufeinanderfolgenden Bildpunkten abgegrenzt. Der erste dargestellte Abschnitt enthält nur Bildpunkte vom Typ b. Dementsprechend wird als Lauflängenkodewort die hexadezimale Zahl 20H gebildet. Im 25 zweiten Abschnitt folgen noch zwei Bildpunkte des Typs b und anschließend sechs Bildpunkte des Typs p. Somit werden hier zwei Lauflängenkodeworte gebildet. Einmal 44H für die beiden Bildpunkte des Typs b und 19H für die sechs aufeinander- folgenden Bildpunkte des Typs p. Im dritten Abschnitt folgen 30 vier Bildpunkte des Typs p und danach vier Bildpunkte des Typs b. Die zugehörigen Lauflängenkodewörter heißen 11H und 10H. Nach diesem erfundungsgemäßem Verfahren wird also eine separate Lauflängenkodierung für eine bestimmte Anzahl von

Bildpunkten durchgeführt. In dem dargestellten Beispiel wurde für jeweils acht Bildpunkte eine separate Lauflängenkodierung gemacht, weil das Beispiel so aufgebaut ist, daß die Laufschrift sich von Schritt zu Schritt um acht Bildpunkte in 5 horizontaler Richtung weiterbewegen soll. Ist in einem anderen Beispiel eine noch feinere Auflösung der Bewegung erforderlich, z.B. lediglich vier Bildpunkte pro Schritt, dann müßten Lauflängenkodewörter jeweils separat für vier Bildpunkte jeder Zeile berechnet werden. Ein Schritt kann 10 dabei bei jedem neuen Videobild erfolgen oder aber auch nur alle N Bilder mit $N = 1, 2, 3, 4, \dots$

In dem erwähnten DVD-Standard ist ein Anzeigesteuerbefehl definiert, mit dem vorgegeben werden kann, bei welcher 15 Speicheradresse die Lauflängendekodierung für das anzuzeigende Teilbild begonnen werden soll. Dieser Befehl ist der SET_DSPXA-Befehl. Es ist dieser Befehl, der in einer erfindungsgemäßen Teilbilddateneinheit eingesetzt wird, um die horizontale Bewegung, z.B. der Laufschrift zu erzeugen. In 20 einer erfindungsgemäßen Teilbilddateneinheit wird dann nämlich für jeden Bewegungsschritt eine eigene Anzeigesteuerbefehlssequenz SP_DCSQ vorgesehen. Jeder dieser Anzeigesteuerbefehlssequenzen verschiebt dabei mittels des SET_DSPXA-Befehls den PXD-Zeiger für den Start der Lauflängendekodierung 25 derart, daß die Lauflängendekodierung um K-Bildpunkte weiter rechts in der Bip-Map beginnt, wobei K die Anzahl der Bildpunkte ist, um die die Laufschrift pro Schritt horizontal laufen soll. Dies ist in Figur 5 durch Pfeile symbolisiert. Bei dem Beispiel gemäß Figur 5 ist K gleich der Anzahl 8. In 30 der ersten Anzeigesteuerbefehlssequenz zeigt der SET_DSPXA-Befehl auf das Lauflängenkodewort 20H. In der nachfolgenden Anzeigesteuerbefehlssequenz zeigt der Befehl auf das Lauflängenkodewort 44H, in der nächsten

Anzeigesteuerungsbefehlssequenz auf das Lauflängenkodewort 11H usw. Die Lauflängendekodierung beginnt demgemäß immer weiter rechts in der dargestellten Zeile. Somit kommen immer andere Bildpunkte der Laufschrift nach und nach zur Anzeige.

5

Ein Beispiel einer erfindungsgemäßen Teilbilddateneinheit für ein DVD-Abspielgerät ist in Figur 6 und 7 angegeben. In der BitMap der Teilbilddateneinheit gemäß Figur 7 sind die Buchstaben A, B, C, D, E, F als ein einziges kombiniertes Bildstück abgelegt. Die Kombinierungsart ist dabei in Figur 6 dargestellt. Im linken Teil der Figur 6 ist das Bildpunkt muster für die Darstellung der Buchstaben A, B, C angegeben. Durch einen einfachen Punkt sind dabei die Hintergrundbildpunkte markiert und durch einen vertikalen Strich die Bildpunkte, die das Muster der Buchstaben erzeugen, markiert. Im mittleren Teil von Figur 6 ist demgemäß das Bildpunkt muster für die Buchstaben D, E, F wiedergegeben. Die Hintergrundbildpunkte sind hier ebenfalls durch einfachen Punkt markiert und die Bildpunkte zur Erzeugung des Buchstabemusters sind in diesem Fall mit einem horizontalen Strich markiert. Erkennbar ist auch die Verschiebung der beiden Teile gegeneinander um zwei Bildpunkte in vertikaler Richtung. Werden die beiden Bildteile so aufeinandergelegt, so entsteht das Bildpunkt muster, wie im rechten Teil der Figur 6 angegeben. Die einfachen Punkte markieren den Bildpunkttyp, wo beide korrespondierende Bildpunkte Hintergrundbildpunkte sind. Der vertikale Strich markiert den Bildpunkttyp, wo der untere Bildpunkt angeschaltet ist und der obere Bildpunkt ausgeschaltet ist. Durch waagerechten Strich sind die Bildpunkte markiert, wo der untere Bildpunkt ausgeschaltet ist, und der korrespondierende obere Bildpunkt angeschaltet ist. Und durch Sternchen sind die Bildpunkte markiert, wo oberer und unterer Bildpunkt beide angeschaltet sind.

Die Lauflängenkodewörter für dieses kombinierte Bildstück sind in dem PXD-Datenfeld der Teilbilddateneinheit abgespeichert. Bei diesem Beispiel sind die Lauflängenkodewörter jeweils für 5 zwei aufeinanderfolgende Pixel separat bestimmt worden. Bei Dekodierung dieser Teilbilddateneinheit ergibt sich dann eine langsame Bewegung des Textes von rechts nach links mit zwei Bildpunkten pro Schritt. Zu beachten ist noch, daß in dem PXD-Feld der Teilbilddateneinheit gemäß Figur 7 verschiedene Bit-10 Map-Bereiche für das erste und zweite Halbbild eines Videobildes abgespeichert sind. Im ersten Halbbild sind somit die Lauflängenkodewörter für die Zeilen 1, 3, 5, 7, 9 und 11 abgespeichert und in der BitMap für das zweite Halbbild sind die Lauflängenkodewörter für die geradzahligen Zeilen 15 abgespeichert. Die Lauflängenkodewörter des kombinierten Bildstückes des ersten Halbbildes stehen ab der Adresse 0004H: im Speicher. In Figur 7 sind die hexadezimalen Adressen ohne den Zusatz des Buchstabens H wiedergegeben. Die Lauflängenkodewörter für das zweite Halbbild stehen ab der 20 Adresse 0033H: im Speicher. Der PXD-Bereich ist ab der Adresse 0060H: mit Null-Byte Einträgen aufgefüllt, damit die im DVD-Standard vorgeschriebene Regel erfüllt ist, daß der PXD-Bereich inklusive des 4-Bytes großen SPU-Kopfbereiches nicht 25 kleiner als die Hälfte der gesamten SPU-Größe sein darf. Nach dem PXD-Bereich folgt der Bereich mit den Anzeigesteuerungsbefehlssequenzen. Dieser Bereich ist auch mit Anzeigesteuerungsbefehlssequenztabelle SP_DCSQT bezeichnet. In der ersten Anzeigesteuerungsbefehlssequenz SP_DCSQ0 geschieht nun das folgende: Mit dem SP_DCSQ_STM-Befehl wird angegeben, 30 daß die Anzeige des Bildes direkt mit dem Aufrufen der Teilbilddateneinheit SPU gestartet werden soll. Der nachfolgende Anzeigesteuerungsbefehl SP_NXT_DCSQ_SA gibt an, wo die nächste Anzeigesteuerungsbefehlssequenz SP_DCSQ1

abgespeichert ist. Durch den nachfolgenden Anzeigesteuerungsbefehl SET_DSPXA wird jetzt festgelegt, daß die Lauflängendekodierung für das erste Halbbild bei der Adresse 004FH: beginnen soll und für das zweite Halbbild bei 5 der Adresse 0033H:. Der darauf folgende Anzeigesteuerungsbefehl SET_COLOR legt fest, daß alle Bildpunkte der Bildpunkttypen b und p als Hintergrund angesehen werden und eine dementsprechende Farbe wird zugeordnet. Außerdem wird mit dem Befehl festgelegt, daß alle 10 Bildpunkte der Bildpunkttypen 1 und 2 als Vordergrundbildpunkte definiert werden und mit einer dementsprechenden anderen Farbe dargestellt werden sollen. Der nachfolgende Befehl SET_CONTR setzt den Kontrast für alle Bildpunkttypen b, p, 1, 2 auf 100% sichtbar, d.h. daß alle 15 Bildpunkte in dem Anzeigefenster für die Teilbilddateneinheit auf voll sichtbar eingestellt sind. Der nächste Befehl SET_DAREA gibt die Größe des sichtbaren Fensters an und definiert den Ort wo das Fenster auf dem Bildschirm erscheinen soll. Im dargestellten Beispiel ist die linke obere Ecke des 20 sichtbaren Fensters bei den Bildkoordinaten (100, 150) und die untere rechte Ecke bei den Bildpunktkoordinaten (117, 159). Mit dem nachfolgenden Befehl STA_DSP wird die Anzeige des ersten Bildes gestartet. Der nächste Befehl CMD_END zeigt das Ende der ersten Anzeigesteuerungsbefehlssequenz SP_DCSQ0 an.

25

Daraufhin wird die zweite Anzeigesteuerungsbefehlssequenz abgearbeitet. In dem SP_DCSQ_STM-Befehl wird jetzt festgelegt, daß die Ausgabe eines neuen Bildes entsprechend dieser Anzeigesteuerungsbefehlssequenz bei dem 30. Bild nach Start 30 der Teilbilddateneinheit angezeigt werden soll. Mit dem nachfolgenden SP_NXT_DCSQ_SA-Befehl wird wieder auf die Speicheradresse hingewiesen, ab der die nächste Anzeigesteuerungsbefehlssequenz SP_DCSQ2 im Speicher abgelegt

ist. Dann folgt der Befehl CHG_COLCON als nächstes. Dieser Befehl dient dazu innerhalb des sichtbaren Fensters festzulegen, welche Bildpunkte als Bildpunkte des unteren Bildstücks in dem kombinierten Bildstück aufzufassen sind und welche als Bildpunkte des oberen Bildstücks in dem kombinierten Bildstück aufzufassen sind. Der CHG_COLCON-Befehl besitzt noch 3 folgende Unterabschnitte. Diese sind mit der Abkürzung LN_CTLI, PX_CTLI und LN_CTLI Termination Code bezeichnet. Das Kodewort für den CHG_COLCON-Befehl besteht nur aus einem Byte, nämlich 07H. Daran anschließend folgt die Angabe wie weit sich der CHG_COLCON-Befehl nachfolgend erstreckt. In der Anzeigesteuерungsbefehlssequenz SP_DCSQ1 ist angegeben, daß noch 16 weitere Bytes zu dem CHG_COLCON-Befehl gehören. Mit dem CHG_COLCON-Befehl können die Farb- und Kontrastwerte für bestimmte Bereiche innerhalb des Anzeigefenster, in dem das Teilbild dargestellt werden soll, geändert werden. Die Angabe LN_CTLI liefert die Information, wieviel Änderungen in den Farb- und Kontrastwerten mit dem Befehl eingestellt werden sollen, dies ist mit der Angabe N = 1 bei dem hier gezeigtem Beispiel lediglich eine Änderung. Dann wird damit auch angegeben, für welchen Bereich auf dem Bildschirm die Änderungen gültig sein sollen. Im gezeigten Beispiel bedeutet die Angabe clsn = 2, daß Änderungen ab der zweite Videozeile gültig sein sollen. Die Angabe ctln = 200 definiert, daß die Änderung bis zur 200. Videozeile gültig sein soll. Praktisch gesehen wirkt sich die Änderung jedoch nur in dem Bereich in dem Anzeigefensters der Teildateneinheit aus. Dieses wurde ja in der ersten Anzeigesteuungsbefehlssequenz SPU_DCSQ0 so definiert, daß es von Pixelposition 100 in der Videozeile 150 bis zur Pixelposition 117 in der Videozeile 159 reicht. Mit der nachfolgenden Information PX_CTLIO wird jetzt angegeben, ab welcher Spalte in dem zuvor definierten Videozeilenbereich

eine Änderung von Farb- und/oder Kontrastwerten erfolgen soll. Die Angabe cspn = 116 definiert, daß ab der 116. Pixelposition eine Änderung erfolgen soll. Die nachfolgende Angabe col = 0 1 0 1 definiert, daß nunmehr für den Pixeltyp b der Farbwert 0 gewählt sein soll, für den Pixeltyp p der Farbwert 1 gültig sein soll, für den Pixeltyp 1 der Farbtyp 0 gültig sein soll und für den Pixeltyp 2 der Farbtyp 1 gültig sein soll. Dies ist verschieden von den Definitionen in der ersten Anzeigesteuerungsbefehlssequenz. Die Angabe CNTR = F F F F definiert, daß für alle Pixeltypen nach wie vor der Kontrast so eingestellt ist, daß die Pixeltypen voll sichtbar sind. Bei den Kontrastwerten ergibt sich somit keine Änderung gegenüber den Angaben in der ersten Anzeigesteuerungsbefehlssequenz. Als letztes folgt noch der Beendigungskode für den CHG_COLCON-Befehl. Mit diesem Befehl wird klargestellt, daß nachfolgend keine weiteren Änderungen einzustellen sind. Durch diesen CHG_COLCON-Befehl ist also erreicht, daß in dem zuvor definierten Anzeigefenster bis zur Spalte 115 einschließlich die Auswertung des kombinierten Bildstücks so erfolgt, wie in der ersten Anzeigesteuerungsbefehlssequenz SP_DCSQ0 angegeben. Ab der Spalte 116 sollen dann die Bildpunkte anders ausgewertet werden. Das bedeutet, daß ab der Spalte 116 statt dem unteren Bildstück in dem kombinierten Bildstück die Bit-Map-Daten so ausgewertet werden, daß das obere Bildstück reproduziert wird. Mit dem letzten Befehl SET_DSPXA in der zweiten Anzeigesteuerungsbefehlssequenz SP_DCSQ1 wird jetzt der Zeiger für den Start der Lauflängenkodierung innerhalb des PXD-Bereiches so gesetzt, daß für das erste Halbbild ab der Adresse 0005H: begonnen werden soll und für das zweite Halbbild ab der Adresse 0034H: begonnen werden soll. Es wird also nicht mehr ab der gleichen Adresse dekodiert wie in der ersten Anzeigesteuerungsbefehlssequenz sondern erst jeweils ein Byte später. Dies ist gleichbedeutend mit der Verschiebung

der im Anzeigefenster ausgegebenen Buchstabenfolge um zwei Bildpunkte nach links, da ja die Lauflängenkodeworte jeweils für zwei aufeinanderfolgende Pixel separat gebildet wurden.

Wie sich die Versetzung des PXD-Zeigers auswirkt, ist anhand

5 der Daten in dem PXD-Feld erkennbar. Das erste Byte unter der Adresse 0004H: wird nämlich so nicht mehr ausgewertet. Es folgen dann 8 Datenbytes mit dem Wert 44H. Dies sind die Kodeworte für jeweils zwei aufeinanderfolgende Hintergrundbildpunkte. Somit werden bei Dekodierung 16

10 Hintergrundbildpunkte in der ersten Zeile ausgegeben. Als nächstes folgt dann das DatenByte 46H. Dieses DatenByte entspricht aber schon dem ersten DatenByte für die dritte Videozeile, wenn ab Adresse 0004H: dekodiert werden würde, wo somit ein Hintergrundbildelement und ein

15 Vordergrundbildelement für das untere Bildstück aufeinanderfolgen. Nach Lauflängendekodierung ergibt sich dann, daß hier ein Bildpunkt vom Typ b auszugeben ist und ein weiterer Bildpunkt vom Typ 1 auszugeben ist. Dadurch, daß diese beiden Typen entsprechend des CHG_COLCON-Befehls beide

20 als Hintergrundbildpunkt aufzufassen sind, werden also an den beiden Pixelpositionen in Spalte 116 und 117 ebenfalls zwei Hintergrundbildpunkte ausgegeben. Zu beachten ist, daß diese letzten beiden Bildpunkte den ersten beiden Bildpunkten des oberen Bildstücks in dem kombinierten Bildstück entsprechen.

25 Dies liegt daran, daß die beiden Bildstücke um zwei Bildpunkte in vertikaler Richtung gegeneinander versetzt wurden. Diese Versetzung garantiert somit, daß immer die richtigen Bildpunkte sich nahtlos aneinanderfügen. Diese Betrachtung ist ebenfalls auch für die Ausgabe des zweiten Halbbildes richtig.

30 In den folgenden Anzeigesteuerbefehlssequenzen wird jetzt entsprechend verfahren um die Laufschrift weiter zu bewegen. Der Bereich, ab dem die Bildpunkte anders auszuwerten sind,

wird sukzessive immer größer gewählt. In der dritten Anzeigesteuerungsbefehlssequenz SP_DCSQ2 wird schon ab der Spalte 114 anders ausgewertet. Pro Anzeigesteuerungsbefehlssequenz wandert der Bereich immer weiter nach links um 5 zwei Bildpunkte. In der neunten Anzeigesteuerungsbefehlssequenz SP_DCSQ8 ist dieser Bereich bereits so groß, daß schon ab der Bildpunktspalte 102 anders ausgewertet wird. Dementsprechend wird auch pro Anzeigesteuerungsbefehlssequenz der PXD-Zeiger um jeweils eine Adresse weiter gerückt. In der 10 neunten Anzeigesteuerungsbefehlssequenz SP_DCSQ8 steht dieser Zeiger für das erste Halbbild bei der Adresse 000CH: und somit bei dem letzten Datenbyte 44H der ersten Videozeile.

In der letzten Anzeigesteuerungsbefehlssequenz SP_DCSQ9 wird 15 dann keine Versetzung des PXD-Zeigers mehr durchgeführt. Ebenfalls findet auch keine Verstellung der Bildpunktauswertung mit dem CHG_COLCON-Befehl mehr statt. Diese Anzeigesteuerungsbefehlssequenz dient lediglich zur Beendigung der Darstellung der Teilbilddateneinheit. Dazu wird 20 der STP_DSP-Befehl benutzt. Zur vollständigen Sichtbarmachung des unteren Bildstücks müßte noch eine zusätzliche Anzeigesteuerungsbefehlssequenz vorgesehen sein, so daß die Umstellung der Auswertung ab Bildpunktspalte 100 beginnt, kombiniert mit der Übersetzung des PXD-Zeigers auf den 25 nächsten Dateneintrag.

Die beschriebene Teilbilddateneinheit produziert somit eine langsame Bewegung der Laufschrift mit den Buchstaben A, B, C, D, E, F, wobei praktisch jede Sekunde um 2 Pixel 30 weitergeschoben wird. Dies liegt daran, daß nur alle 30 Videobilder eine Versetzung erfolgt. Natürlich könnte diese Bewegung beschleunigt werden, wenn die Änderungen in kürzeren Abständen zur Anzeige gebracht werden sollen.

Ebenfalls läßt sich auf die gleiche Art und Weise sehr einfach die Laufrichtung für die Laufschrift auch umkehren. In dem Fall müßte der PXD-Zeiger in umgekehrter Richtung versetzt werden und der CHG_COLCON-Befehl entsprechend angepaßt werden, so, daß der Bereich wo anders ausgewertet werden soll sukzessive von links nach rechts vergrößert wird. In dem gezeigten Beispiel wurde eine nur sehr kurze Laufschrift der Einfachheit halber erzeugt. Für längere Laufschriften muß jetzt noch das folgende beachtet werden.

Dazu wird auf die Figur 8 verwiesen. Dort ist eine längere Laufschrift dargestellt, die von dem Buchstaben a bis zum Buchstaben o reicht. Angezeigt werden sollen von dieser Laufschrift jeweils immer drei Buchstaben in dem entsprechend gewählten Anzeigefenster. Dann muß die gesamte Laufschrift mit vertikalen Schnitten in gleich große Bereiche eingeteilt werden, von denen jeder immer drei Buchstaben umfaßt. Zwei aufeinanderfolgende Teile werden dann miteinander kombiniert und in einer zugehörigen Bit-Map im Speicher abgelegt. In Figur 8 ist dies durch geschweifte Klammer angedeutet. Es entsteht ein erstes kombiniertes Bildstück mit einer dementsprechenden ersten Bit-Map. Darin sind die Buchstaben a, b, c und d, e, f kombiniert abgelegt. In dem zweiten kombinierten Bildstück sind dann die Buchstaben d, e, f und g, h, i kombiniert worden usw. Die kombinierten Bildstücke können im PXD-Bereich hintereinander im Speicher eingetragen werden. Wurde das erste kombinierte Bildstück mit der ersten Bit-Map komplett abgearbeitet, wie zuvor erläutert, so daß bei einer Bewegung von rechts nach links nur noch Bildpunkte des zweiten Bildstücks sichtbar sind, dann wird die Darstellung des nächsten kombinierten Bildstückes begonnen. Es wird dann also die zweite BitMap in einer folgenden Anzeigesteuierungs-

- befehlssequenz aufgerufen. Ab hier wird dann wie bei den ersten beiden Bildstücken verfahren. Mit den restlichen kombinierten Bildstücken wird analog verfahren bis das letzte kombinierte Bildstück abgearbeitet wurde. Mit der
- 5 beschriebenen Methode ist ebenfalls auch ein variierendes horizontales Durchlaufen (z.B. Richtungs-wechsel, variierende Laufgeschwindigkeiten, usw.) bei entsprechend angepaßter Lauflängenkodierung einfach realisierbar.
- 10 Bei dem hier erläuterten Beispiel wurde davon ausgegangen, daß die kombinierten Bildstücke in zwei Halbbildern separiert im PXD-Bereich abgespeichert sind und über den SET_DSPXA-Befehl separat angesprochen werden. Die Methode ist jedoch ebenso für die Erzeugung einer horizontalen Bewegung geeignet, bei denen
- 15 der SET_DSPXA-Befehl für beide Halbbilder auf dieselbe Speicheradresse in dem PXD-Bereich gesetzt wird. Dieses Verfahren wird auch angewendet und ist unter dem Begriff "field repetition" bekannt.
- 20 Das erwähnte Kodierverfahren zur Kodierung einer Bildsequenz kann noch weiter verfeinert werden. Z.B. kann die Anordnung der kombinierten Bildstücke noch optimiert werden. Bei dem Beispiel gemäß Figur 8 stimmt nämlich z.B. bei einer Bewegung von rechts nach links das ausgegebene Bild basierend auf dem
- 25 kombinierten Bildstück K1 bei der ganz nach links verschobenen Ausgabeposition mit dem ausgegebenen Bild basierend auf dem kombinierten Bildstück K2 bei der ganz nach rechts geschobenen Ausgabeposition überein und ist damit redundant vorhanden. Soll somit erreicht werden, daß beim Übergang von einem
- 30 kombinierten Bildstück zum darauffolgenden das Bild ebenfalls um die definierte Anzahl von Bildpunkten von rechts nach links bewegt wird, dann kann das so erreicht werden, daß das zweite kombinierte Bildstück um die entsprechende Anzahl von

Bildpunkten nach rechts versetzt wird gegenüber dem vorigen Bildstück. Dies ist in der Figur 9 dargestellt. Die kombinierten Bildstücke K1 und K2, wie auch K2 und K3 und K3 und K4 sind jeweils um 8 Bildpunkte zusätzlich gegeneinander 5 verschoben verglichen zu der Darstellung gemäß Figur 8. Dies hat zusätzlich den Vorteil, daß dadurch entweder insgesamt weniger kombinierte Bildstücke benötigt werden und somit weniger Speicher für den PXD-Bereich benötigt wird oder im letzten kombinierten Bildstück sind weniger 10 Vordergrundbildpunkte enthalten, was wiederum eine den Speicherbedarf reduzierende Lauflängenkodierung erlaubt. Dies ist in Figur 10 veranschaulicht.

Im oberen Teil der Figur 10 sind die Bildstücke so kombiniert, 15 wie in Figur 8. Für die gesamte Laufschrift, entsprechend der durchgezogenen Linie, sind somit acht kombinierte Bildstücke erforderlich. Im unteren Teil der Figur 10 wurde so kombiniert, wie in Figur 9 dargestellt. Erkennbar ist, daß nur sieben kombinierte Bildstücke K1 - K7 abgespeichert werden 20 müssen. Wie auch schon zuvor erläutert, ist ebenfalls dargestellt, daß das letzte Bildstück mit Hintergrundbildpunkten aufgefüllt wird, wenn das letzte Bildstück nicht die gleiche Größe wie das vorhergehende Bildstück aufweist, damit eine Kombination überhaupt möglich 25 wird. Dieser Bereich ist gestrichelt dargestellt. Dadurch ergeben sich im letzten kombinierten Bildstück K7 ungenutzte Bildbereiche im oberen Bildstück. Daher wird für diesen ungenutzten Bereich keine besondere Lauflängenkodierung mehr benötigt, d.h. die Lauflängenkodierung des ungenutzten 30 Bereiches kann ganz normal kodiert werden, also rein speicherminimierend. Einzige Einschränkung: Das spezielle Lauflängenkodewort "Till-end-of-line" darf nicht verwendet werden, da sonst eine fehlerhafte Kodierung stattfinden würde,

weil ja der PXD-Zeiger sukzessive immer weiter versetzt wird und daher das Zeilenende ständig wechselt.

- Eine weitere Speicherbedarfsreduzierung kann erreicht werden,
- 5 wenn die letzten Zeilen eines Objektes z.B. einer Laufschrift einfacher lauflängenkodiert werden. Dieses ist in Figur 11 veranschaulicht. Enthalten die unteren Zeilen des Objektbildes nur einen Bildpunkttyp (z.B. nur Hintergrundbildpunkte), dann kann die Lauflängenkodierung dieses Bereiches stark
- 10 vereinfacht werden. Sämtliche letzten Zeilen mit nur einem Bildpunkttyp eines jeden Halbbildes werden ausschließlich mit dem Lauflängenkodewort "Till-end-of-line" kodiert. Außerdem wird jedem Halbbild noch ein "Till-end-of-line" Kodewort angehängt. Im oberen Teil der Figur 11 ist die bisher
- 15 erläuterte Methode für die Lauflängenkodierung schematisch angegeben. Danach werden alle Zeilen gleich behandelt und zugehörige Lauflängenkodewörter für alle Ausschnitte separat erzeugt. Dies gilt auch für die letzten Zeilen, wo nur noch Hintergrundbildpunkte enthalten sind. Im unteren Teil der
- 20 Figur 11 ist dann erkennbar, daß die letzten Zeilen, markiert durch geschweifte Klammer und das Bezugszeichen TEOL mit den Kodewörtern "Till-end-of-line" kodiert sind. In diesen Zeilen tritt kein Vordergrundbildpunkt mehr auf.
- 25 Das erfindungsgemäße Verfahren zur Kodierung einer Bildsequenz und dementsprechend auch die erfindungsgemäßen Teilbilddateneinheiten können in all den Geräten eingesetzt werden, die den erwähnten Sub-Picture-Dekoder entsprechend des DVD-Standards als Komponente enthalten. Dies sind also
- 30 insbesondere DVD-Geräte inklusive DVD-RAM und DVD-Recordable sowie auch gegebenenfalls in Fernsehempfangsgeräten wie Set-Top-Boxen oder auch beispielsweise in Camcordern.

Ansprüche

1. Verfahren zur Kodierung einer Bildsequenz, bei der ein Bild jeweils einen Ausschnitt eines Objektes darstellt, wobei bei Darstellung der Bildsequenz in einem Anzeigefenster auf einem Bildschirm für den Betrachter der Eindruck einer horizontalen Bewegung des Objektes in dem Anzeigefenster entsteht, **dadurch gekennzeichnet, daß** durch vertikale Schnitte das Objektbild in eine Anzahl gleichgroßer Segmente (30) zerlegt wird, daß die Bildpunktdata von jeweils zwei zusammenhängenden Segmenten (30) kombiniert werden, wobei jeweils aus korrespondierenden Bildpunkten der zu kombinierenden Segmente ein kombinierter Bildpunkt erzeugt wird, derart, daß dem kombinierten Bildpunkt ein eindeutiger Typ (b, p, 1, 2) zugewiesen wird, der der Verteilung der Bildpunktwerte in dem Paar der korrespondierenden Bildpunkte entspricht, und das eine definierte Anzahl von kombinierten Bildpunkten pro Zeile des Objektbildes jeweils separat lauflängenkodiert wird, wobei die definierte Anzahl von kombinierten Bildpunkten zu der Anzahl von Bildpunkten korrespondiert, um die das Objektbild pro Bewegungsschritt in horizontaler Richtung bewegt wird.
2. Verfahren nach Anspruch 1, wobei die Größe der Segmente (30) zu der Größe der darzustellenden Objektausschnitte korrespondiert.
3. Verfahren nach Anspruch 1 oder 2, wobei die jeweils zwei zusammenhängenden Segmente gegeneinander um zwei Bildpunkte in vertikaler Richtung verschoben werden bevor sie kombiniert werden.

4. Verfahren nach Anspruch 1 oder 2, wobei die jeweils zwei zusammenhängenden Segmente gegeneinander um einen Bildpunkt in vertikaler Richtung verschoben werden bevor sie kombiniert werden.
5
5. Verfahren nach einem der Ansprüche 1 bis 4, wobei ein kombiniertes Bildstück gegenüber dem vorhergehenden und nachfolgenden Bildstück um die Anzahl Bildpunkte um die das Objekt pro Bild in horizontaler Richtung bewegt wird, versetzt ist.
10
6. Verfahren nach einem der Ansprüche 1 bis 5, wobei die Bildpunkte der Bilder nur zwei Werte annehmen können, nachfolgend Wert "An" und "Aus" genannt.
15
7. Verfahren nach einem der Ansprüche 1 bis 6, wobei die folgenden vier unterschiedlichen Bildpunkttypen vorgesehen sind:
20
 - haben die korrespondierenden Bildpunkte beide den Wert "Aus", so wird dem kombinierten Bildpunktwert der Typ A zugeordnet;
 - haben die korrespondierenden Bildpunkte beide den Wert "An", so wird dem kombinierten Bildpunktwert der Typ B zugeordnet;
25
 - hat der Bildpunkt des ersten Bildsegmentes den Wert "An" und der Bildpunkt des zweiten des zu kombinierenden Bildsegmentes den Wert "Aus", so wird dem kombinierten Bildpunktwert der Typ C zugeordnet;
 - hat der Bildpunkt des ersten der beiden zu kombinierenden Bildsegmente den Wert "Aus" und der Bildpunkt des zweiten der beiden zu kombinierenden
30

Bildsegmente den Wert "An", so wird dem kombinierten Bildpunktwert der Typ D zugeordnet.

8. Verfahren nach einem der Ansprüche 1 bis 7, wobei die zu kombinierende Bildsegmente nur teilweise überlappend kombiniert werden und fehlende Bildpunkte außerhalb des Überlappungsbereichs den Wert "Aus" zugewiesen bekommen.
9. Teilbilddateneinheit zur Verwendung in einem elektronischen Gerät, insbesondere Unterhaltungselektronikgerät, wobei die Teilbilddateneinheit eine Bildsequenz enthält, die nach einem Verfahren nach einem der Ansprüche 1 bis 7 kodiert wurde, mit einem Datenbereich (21) für die Bildpunktwerte der Bilder der Bildsequenz und mit einem Befehlsbereich (22) für Anzeigesteuerungsbefehlssequenzen (SP_DCSQ), **dadurch gekennzeichnet, daß** die Anzeigesteuerungsbefehlssequenzen (SP_DCSQ) einen Befehl beinhalten, der den Zeiger für die Startadresse der Lauflängendekodierung jeweils so setzt, daß die Lauflängendekodierung anschließend bei anderen Bildpunkten des Objektbildes beginnt, derart, daß gegenüber der vorherigen Lauflängendekodierung der Start der Lauflängendekodierung um soviele Bildpunkte nach rechts oder links versetzt ist, wie das Objektbild entsprechend der gewünschten Bewegung verschoben werden soll und, daß die Anzeigesteuerungsbefehlssequenzen einen Befehl beinhalten, der festlegt, auf welche Weise die kombinierten Bildpunkte eines kombinierten Bildstücks in zugehörigen Bereichen auszuwerten sind.

10. Teilbilddateneinheiten nach Anspruch 9, wobei die Teilbilddateneinheit für die Verwendung in einem DVD-Gerät ausgelegt ist.

5 11. Teilbilddateneinheit nach Anspruch 10, wobei die Anzeigesteuerungsbefehlssequenzen (SP_DCSQ) den in dem DVD-Standard festgelegten Anzeigesteuerungsbefehl SET_DSPXA enthalten, um den Lauflängendekodierungszeiger zu versetzen.

10 12. Teilbilddateneinheit nach Anspruch 10 oder 11, wobei jedem der in Anspruch 6 definierten Bildpunkttypen A, B, C, D in mathematisch eindeutiger Weise einer der im DVD-Standard festgelegten Bildpunkttypen "Background-Pixel",
15 "Pattern-Pixel", "Emphasis-1-Pixel" und "Emphasis-2-Pixel" zugeordnet wird.

20 13. Teilbilddateneinheit nach einem der Ansprüche 10 bis 12, wobei in einer Anzeigesteuerungsbefehlssequenz (SP_DCSQT), der in dem DVD-Standard definierte Anzeigesteuerungsbefehl SET_COLOR oder CHG_COLCON benutzt wird, um festzulegen, auf welche Weise die kombinierten Bildpunkte eines kombinierten Bildstücks in zugehörigen Bereichen auszuwerten sind.

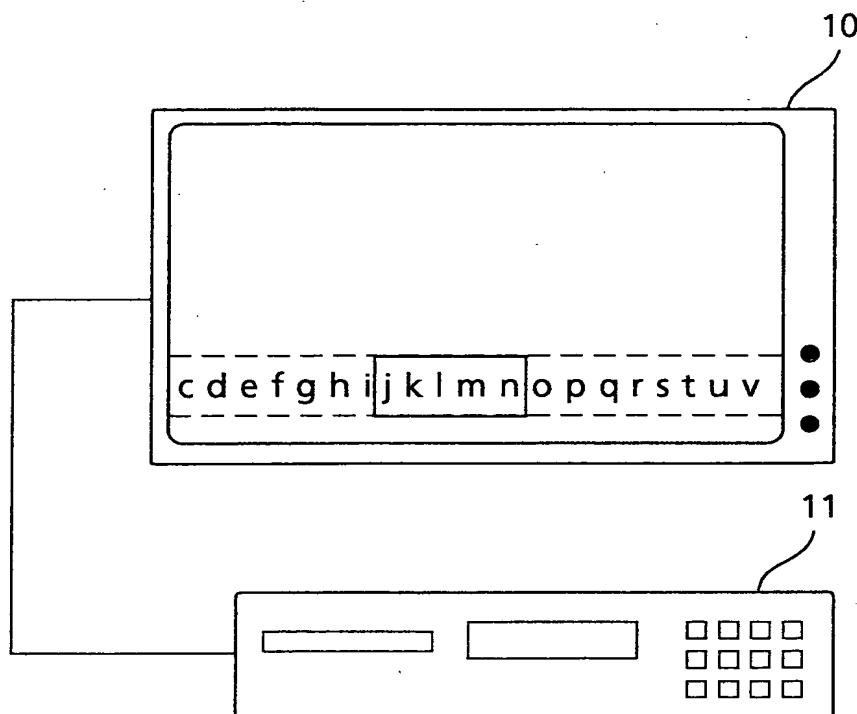
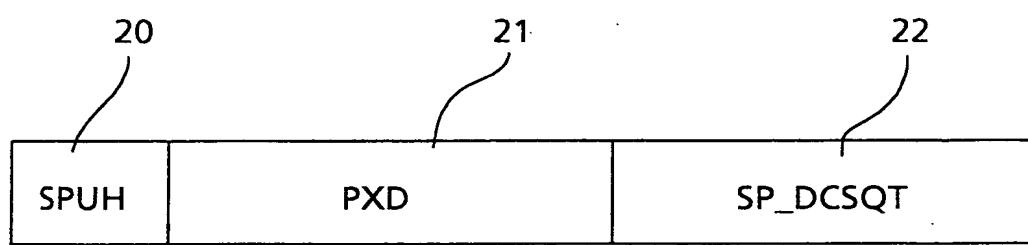
25 14. Datenträger, insbesondere DVD-Scheibe, aufweisend eine Teilbilddateneinheit nach einem der Ansprüche 9 bis 13.

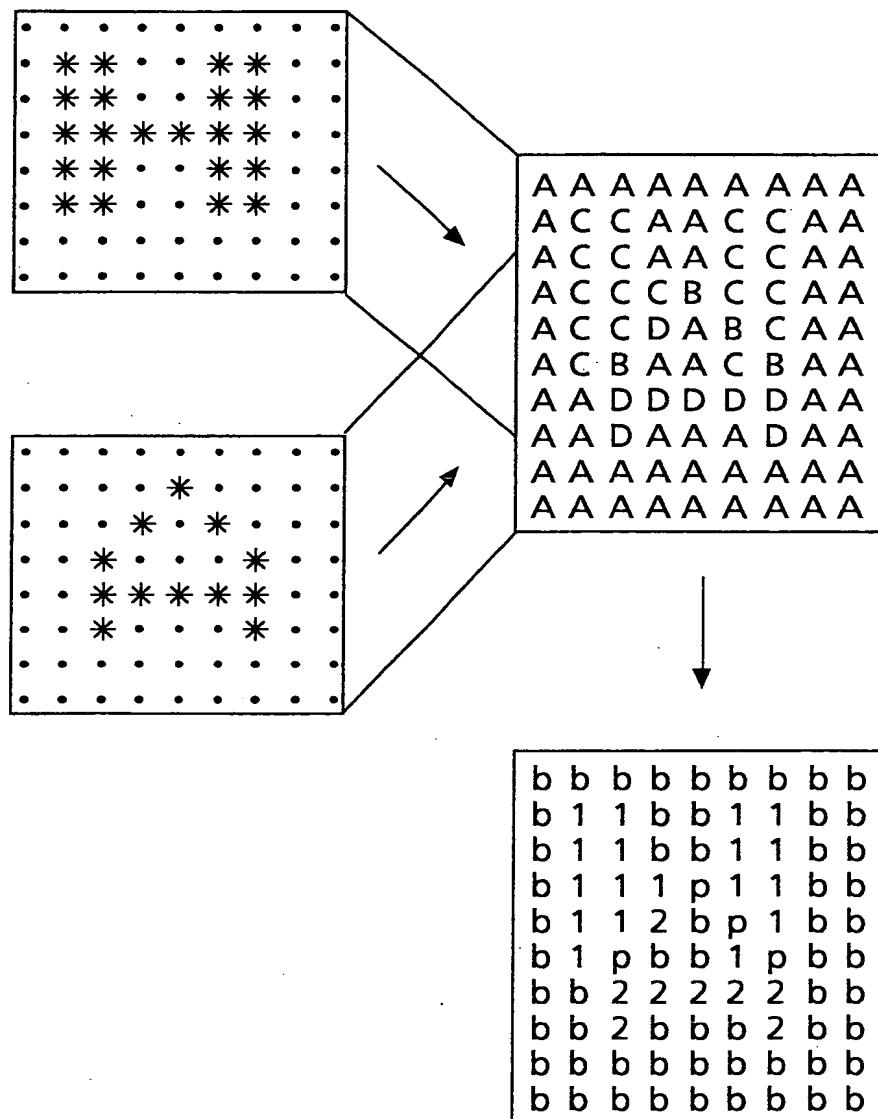
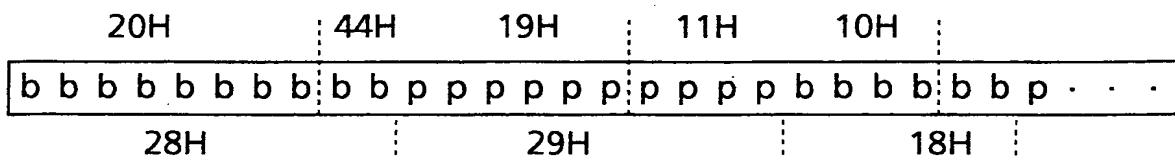
Zusammenfassung

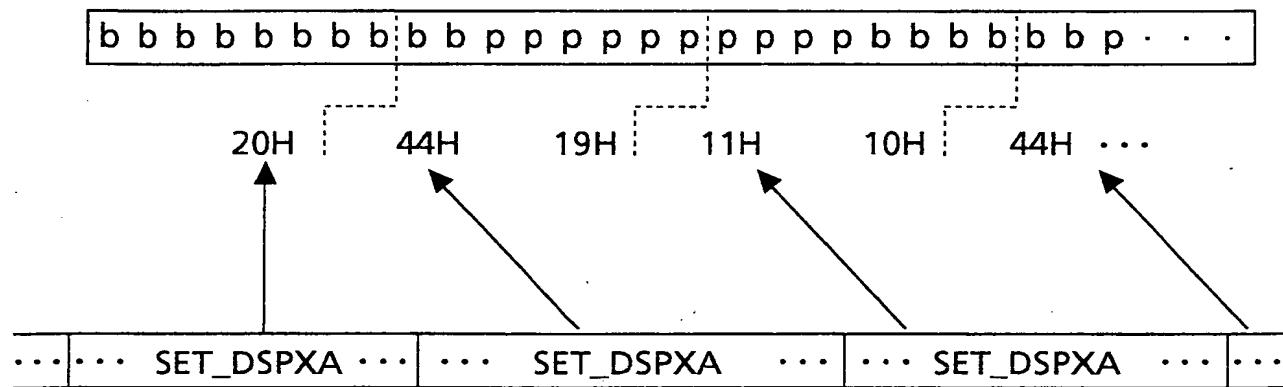
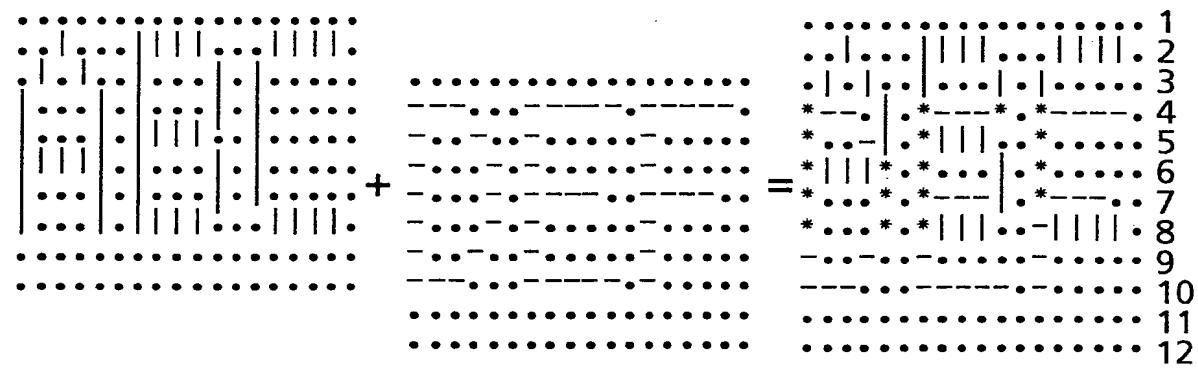
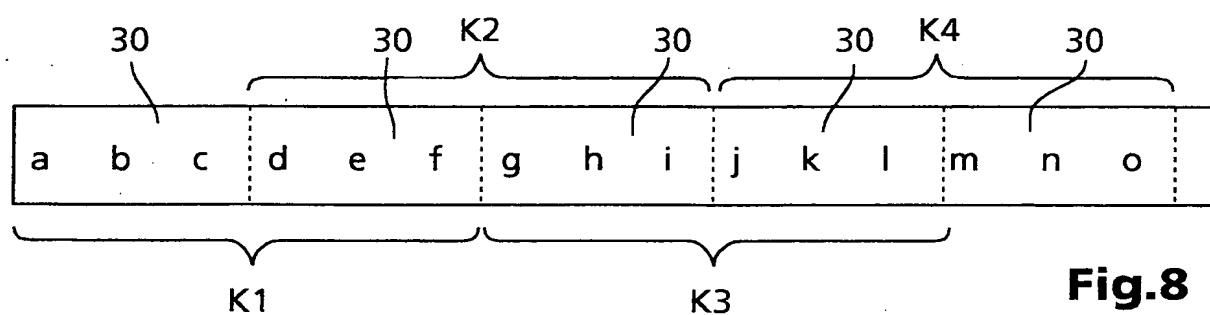
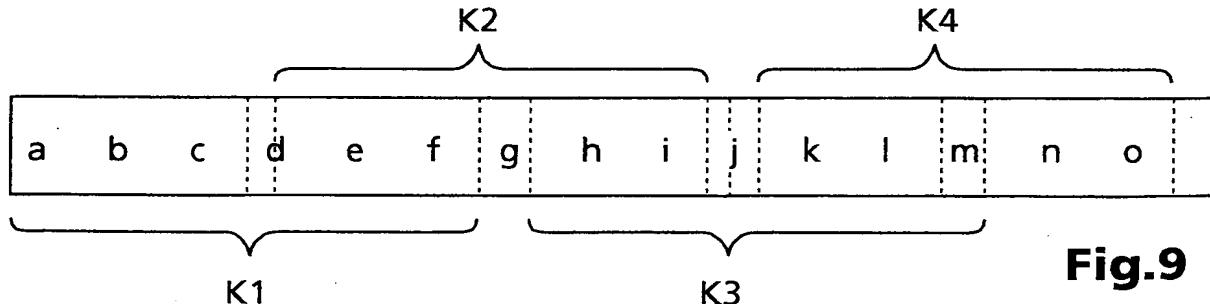
Im DVD-Standard ist spezifiziert wie auf einfache Weise z.B. Untertitel in ein Videobild eingeblendet werden können. Pro 5 Einblendung ist danach jeweils eine sogenannte Teilbilddateneinheit auf einer DVD-Scheibe vorzusehen. Hinweise, wie mit solchen Teilbilddateneinheiten auch horizontal bewegte Objekte in einem Anzeigefenster dargestellt werden können, z.B. für die Realisierung einer Laufschrift, 10 sind in dem DVD-Standard nicht enthalten. Die Erfindung setzt sich zum Ziel solche horizontal bewegten Objekte wie Laufschriften mit den im DVD-Standard spezifizierten Regeln und Befehlen speicherbedarfsoptimiert erstellen zu können. Die Erfindung löst die Aufgabe durch die Verwendung einer 15 speziellen Lauflängenkodierungsart in Verbindung mit einer besonderen Kombinierungsart von Objektbildsegmenten. Die Erfindung betrifft ein dementsprechendes Verfahren zur Kodierung einer Bildsequenz und eine dementsprechende Teilbilddateneinheit sowie einen Datenträger.

20

Fig. 3

**Fig.1****Fig.2**

**Fig.3****Fig.4**

**Fig.5****Fig.6****Fig.8****Fig.9**

4/6

0000: ##### SPUH:
 0000: 01 FC # SPDSZ = 508 = 0x1FC
 0002: 00 FE # SP_DCSQTA = 254 = 0x FE

0004: ##### PXD:
 0004: 44 44 44 44 44 44 44 44 44 44 44 46 46 46 44 64 44 64 64 64
 0014: 44 44 74 45 64 76 66 44 74 44 44 74 44 74 44 74 75 55
 0024: 64 75 55 44 54 45 44 54 44 44 54 44 44 44 44 00 00 00 44
 0034: 64 44 66 66 44 46 66 64 75 54 64 75 55 74 75 55
 0044: 54 76 66 74 74 44 64 74 44 44 74 44 74 74 76 66 44
 0054: 56 66 64 55 54 44 55 55 54 54 44 44 00 00 00 00 00 00 00
 0064: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0074: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0084: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0094: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00A4: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00B4: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00C4: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00D4: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00E4: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00F4: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

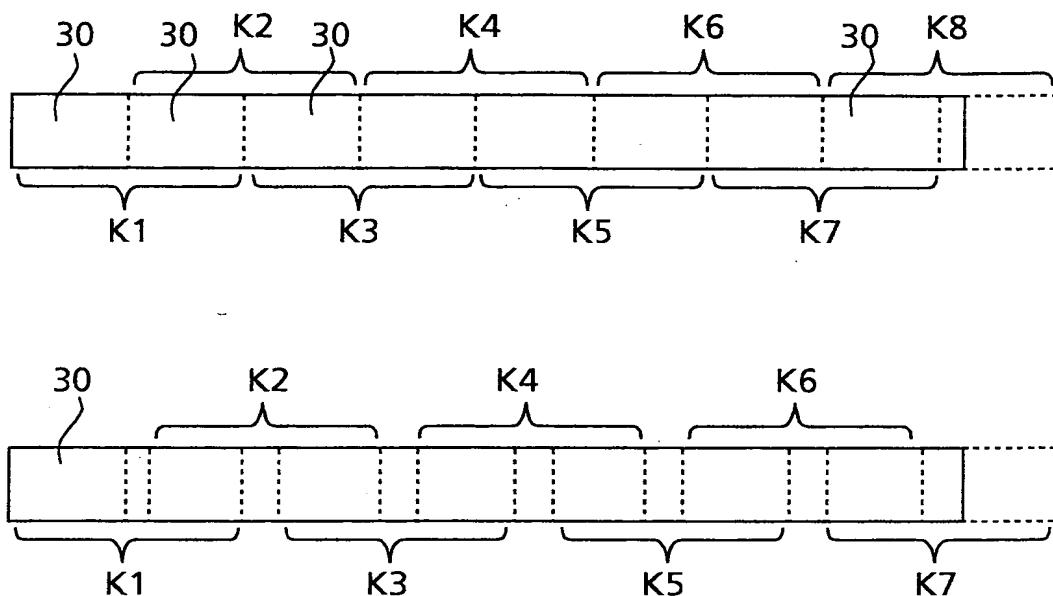
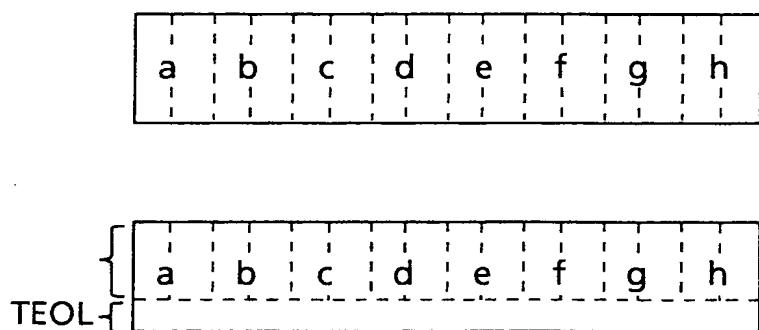
00FE: ##### SP_DCSQT:
 00FE: ##### SP_DCSQ 0 = 0x0
 00FE: 00 00 # SP_DCSQ_STM = 0s => start frame: 0
 0100: 01 16 # SP_NXT_DCSQ_SA = 278 = 0x116
 0102: 06 00 04 00 33 # SET_DSPXA uf=0x4 bf=0x33
 0107: 03 11 00 # SET_COLOR b=0 p=0 e1=1 e2=1
 010A: 04 FF FF # SET_CONTR (0..15) b=15 p=15 e1=15 e2=15
 010D: 05 06 40 75 09 60 9F # SET_DAREA sx=100 ex=117 sy=150 ey=159
 0114: 01 # STA_DSP
 0115: FF # CMD_END
 0116: ##### SP_DCSQ 1 = 0x1
 0116: 00 57 # SP_DCSQ_STM => start frame: 30
 0118: 01 31 # SP_NXT_DCSQ_SA = 305 = 0x131
 011A: 07 00 10 # CHG_COLCON (Extended field size = 16 = 0x10)
 011D: 00 02 10 C8 # LN_CTLI 0: n=1 csln=2 ctln=200
 0121: 00 74 10 10 FF FF # PX_CTLI 0: cspn=116 col= 0 1 0 1 cntr=F F F F
 0127: 0F FF FF FF # LN_CTLI Termination Code
 012B: 06 00 05 00 34 # SET_DSPXA uf=0x5 bf=0x34
 0130: FF # CMD_END
 0131: ##### SP_DCSQ 2 = 0x2
 0131: 00 AF # SP_DCSQ_STM => start frame: 60
 0133: 01 4C # SP_NXT_DCSQ_SA = 332 = 0x14C
 0135: 07 00 10 # CHG_COLON (Extended field size = 16 = 0x10)
 0138: 00 02 10 C8 # LN_CTLI 0: n=1 csln=2 ctln=200
 013C: 00 72 10 10 FF FF # PX_CTLI 0: cspn=114 col= 0 1 0 1 cntr=F F F F
 0142: 0F FF FF FF # LN_CTLI Termination Code
 0146: 06 00 06 00 35 # SET_DSPXA uf=0x6 bf=0x35
 014B: FF # CMD_END
 014C: ##### SP_DCSQ 3 = 0x3
 014C: 01 07 # SP_DCSQ_STM => start frame: 90
 014E: 01 67 # SP_NXT_DCSQ_SA = 359 = 0x167
 0150: 07 00 10 # CHG_COLCON (Extended field size = 16 = 0x10)
 0153: 00 02 10 C8 # LN_CTLI 0: n=1 csln=2 ctln=200
 0157: 00 70 10 10 FF FF # PX_CTLI 0: cspn=112 col= 0 1 0 1 cntr=F F F F
 015D: 0F FF FF FF # LN_CTLI Termination Code
 0161: 06 00 07 00 36 # SET_DSPXA uf=0x7 bf=0x36
 0166: FF # CMD_END

```

0167: ##### SP_DCSQ 4 = 0x4
0167: 01 5F      # SP_DCSQ_STM => start frame: 120
0169: 01 82      # SP_NXT_DCSQ_SA = 386 = 0x182
016B: 07 00 10    # CHG_COLON (Extended field size = 16 = 0x10)
016E: 00 02 10 C8 # LN_CTLI 0: n=1 csln=2 ctln=200
0172: 00 6E 10 10 FF FF # PX_CTLI 0: cspn=110 col= 0 1 0 1 cntr=F F F F
0178: 0F FF FF FF # LN_CTLI Termination Code
017C: 06 00 08 00 37 # SET_DSPXA uf=0x8 bf=0x37
0181: FF          # CMD_END
0182: ##### SP_DCSQ 5 = 0x5
0182: 01 B7      # SP_DCSQ_STM => start frame: 150
0184: 01 9D      # SP_NXT_DCSQ_SA = 413 = 0x19D
0186: 07 00 10    # CHG_COLCON (Extended field size = 16 = 0x10)
0189: 00 02 10 C8 # LN_CTLI 0: n=1 csln=2 ctln=200
018D: 00 6C 10 10 FF FF # PX_CTLI 0: cspn=108 col= 0 1 0 1 cntr=F F F F

0193: 0F FF FF FF # LN_CTLI Termination Code
0197: 06 00 09 00 38 # SET_DSPXA uf=0x9 bf=0x38
019C: FF          # CMD_END
019D: ##### SP_DCSQ 6 = 0x6
019D: 02 0F      # SP_DCSQ_STM => start frame: 180
019F: 01 B8      # SP_NXT_DCSQ_SA = 440 = 0x1B8
01A1: 07 00 10    # CHG_COLCON (Extended field size = 16 = 0x10)
01A4: 00 02 10 C8 # LN_CTLI 0: n=1 csln=2 ctln=200
01A8: 00 6A 10 10 FF FF # PX_CTLI 0: cspn=106 col= 0 1 0 1 cntr=F F F F
01AE: 0F FF FF FF # LN_CTLI Termination Code
01B2: 06 00 0A 00 39 # SET_DSPXA uf=0xA bf=0x39
01B7: FF          # CMD_END
01B8: ##### SP_DCSQ 7 = 0x7
01B8: 02 67      # SP_DCSQ_STM => start frame: 210
01BA: 01 D3      # SP_NXT_DCSQ_SA = 467 = 0x1D3
01BC: 07 00 10    # CHG_COLON (Extended field size = 16 = 0x10)
01BF: 00 02 10 C8 # LN_CTLI 0: n=1 csln=2 ctln=200
01C3: 00 68 10 10 FF FF # PX_CTLI 0: cspn=104 col= 0 1 0 1 cntr=F F F F
01C9: 0F FF FF FF # LN_CTLI Termination Code
01CD: 06 00 0B 00 3A # SET_DSPXA uf=0xB bf=0x3A
01D2: FF          # CMD_END
01D3: ##### SP_DCSQ 8 = 0x8
01D3: 02 BF      # SP_DCSQ_STM => start frame: 240
01D5: 01 EE      # SP_NXT_DCSQ_SA = 494 = 0x1EE
01D7: 07 00 10    # CHG_COLCON (Extended field size = 16 = 0x10)
01DA: 00 02 10 C8 # LN_CTLI 0: n=1 csln=2 ctln=200
01DE: 00 66 10 10 FF FF # PX_CTLI 0: cspn=102 col= 0 1 0 1 cntr=F F F F
01E4: 0F FF FF FF # LN_CTLI Termination Code
01E8: 06 00 0C 00 3B # SET_DSPXA uf=0xC bf=0x3B
01ED: FF          # CMD_END
01EE: ##### SP_DCSQ 9 = 0x9
01EE: 03 17      # SP_DCSQ_STM => start frame: 270
01F0: 01 EE      # SP_NXT_DCSQ_SA = 494 = 0x1EE
01F2: 07 00 06    # CHG_COLON (Extended field size = 6 = 0x6)
01F5: 0F FF FF FF # LN_CTLI Termination Code
01F9: 02          # STP_DSP
01FA: FF          # CMD_END
01FB: FF          # Stuffing Byte => even SP size

```

**Fig.10****Fig.11**